



22



Múltiplas contas GIT no mesmo computador

 Ouvir

Ping!

Por Allan Monteiro Última atualização 9 meses atrás ⌚ 5 Min. 🚩

Este artigo é um guia prático para configurar e gerenciar várias contas do GitHub em uma única máquina, portanto, presumo que você também envolva o processo.



1



Próximo >



Em primeiro lugar, você deve ter duas contas GitHub/Bitbucket (entre outras) ativas diferentes. Vamos supor que sua conta pessoal do GIT esteja relacionada a `personal@email.com` e sua conta do GIT do escritório esteja associada a `office@email.com`

- **Crie chaves SSH para duas contas GitHub**

Em seguida, temos que gerar duas chaves SSH para contas GIT correspondentes. O comando a seguir criará uma chave para sua conta pessoal.

Javascript

```
cd ~/.ssh  
ssh-keygen -t rsa -b 4096 -C "personal@email.com"
```

Isso pedirá um nome para salvar a chave. Digite `id_rsa_personal` para salvá-lo. Depois que ele solicitar uma senha, deixe em branco e vá em frente.

O seguinte comando criará uma chave para sua conta de escritório. Quando ele pedir um nome para salvar a chave, digite `id_rsa_office`, deixe a senha vazia como no passo anterior.

Javascript

```
ssh-keygen -t rsa -b 4096 -C "office@email.com"
```

Depois de executar com sucesso o comando acima irá criar quatro arquivos dentro do diretório `.ssh`. Você pode listá-los executando este comando:

```
ls -al ~/.ssh
```

4 arquivos gerados serão parecidos com este

```
id_rsa_personal  
id_rsa_personal.pub  
id_rsa_office  
id_rsa_office.pub
```

Agora você tem duas chaves. Um para sua conta pessoal e outro para sua conta do escritório. Observe que os arquivos com extensão .pub são arquivos que precisamos adicionar nas contas do GitHub. O próximo passo é para isso.

- **Adicionar chaves SSH a contas GitHub**

Copie a chave pública da conta pessoal usando o seguinte comando. Isso copiará a chave para a área de transferência.

```
pbcopy < ~/.ssh/id_rsa_personal.pub
```

Vá para Git provider que está configurando (GitHub, etc) e faça login em sua conta pessoal (personal@email.com). Siga esses passos.

1. Clique em seu avatar no canto superior direito e clique no menu Configurações.
2. No lado esquerdo, no menu Configurações pessoais, clique nas chaves SSH e GPG.
3. No canto superior direito, clique no botão Nova chave SSH.
4. Adicione um título significativo para identificar sua chave.
5. Pressione cmd + v para colar a chave copiada.
6. Clique em Adicionar chave SSH para adicionar sua chave e digite sua senha se for solicitada.

Siga as mesmas etapas para adicionar sua chave SSH do escritório.

- **Crie um arquivo de configuração para adicionar duas chaves**

Usando o seu terminal, volte para a pasta .ssh digitando cd ~ / .ssh crie um arquivo de configuração usando o comando abaixo.

```
touch config
```

Javascript

Abra e edite o arquivo usando o nano (ou qualquer editor de sua preferência).

```
nano config
```

Javascript

Em seguida, adicione as seguintes configurações para duas contas.

```
# Personal account
Host github.com-personal
  HostName github.com
  User git
  IdentityFile ~/.ssh/id_rsa_personal

# Office account
Host github.com-office
  HostName github.com
  User git
  IdentityFile ~/.ssh/id_rsa_office
```

Para salvar e sair, pressione ctrl + o, depois pressione Enter e depois pressione ctrl + x

- **Criar arquivos de configuração git**

Usar dois arquivos de configuração git para nossas contas pessoais e de escritório reduzirá o estresse de gerenciar essas contas, automatizando o processo manual. Criaremos um arquivo de configuração git globalmente para nossa conta pessoal e o outro em nossa pasta de trabalho do Office.

Usando seu terminal, navegue até a pasta raiz para criar um arquivo de configuração git global para sua conta pessoal.

```
cd ~
nano .gitconfig
```

Em seguida, adicione as seguintes configurações.

Javascript

```
[user]
  name = Your Name
  email = personal@email.com

[includeIf "gitdir:~/office/"]

  path = ~/office/.gitconfig
```

Observe que para trabalhar com a configuração de escritório acima, git assume que você tem todas as coisas relacionadas ao trabalho de escritório localizadas em uma pasta chamada Office. Portanto, se não for o seu caso, sinta-se à vontade para alterar a configuração acima de acordo com a estrutura de sua pasta.

Em seguida, crie um arquivo de configuração git relacionado ao trabalho de escritório. Para fazer isso, navegue até a pasta Office e digite os comandos a seguir.

Javascript

```
nano .gitconfig
```

Em seguida, adicione as seguintes configurações.

Javascript

```
[user]

  email = office@email.com
```

- **Salvar identidades-chave na máquina local**

Primeiro, temos que remover todas as identidades existentes executando o seguinte comando.

```
cd ~  
ssh-add -D
```

Javascript

Após a exclusão, verifique se as chaves recém-adicionadas foram salvas com sucesso executando o seguinte comando.

```
ssh-add -l
```

Javascript

Agora você precisa autenticar as chaves com o GitHub executando os seguintes comandos.

```
ssh -T github.com-personal  
ssh -T github.com-office
```

Javascript

Agora, estamos quase terminando!





- A etapa final, teste usando o repositório GitHub

Vá para sua conta pessoal do GitHub e você pode criar um novo repositório lá ou usar um existente. Vamos supor que o nome do seu repositório seja my-repo. Siga as instruções abaixo e faça um clone de seu repositório em sua máquina local.

```
git clone git@github.com:personal:USERNAME/my-repo.git
```

Javascript

Isso clonará seu repositório e o resto é história. Você sabe o que fazer, vá em frente e crie o commit inicial! Aplique o mesmo procedimento para contas de escritório. Tenha em mente que, para ter sucesso neste método, o git assume que todas as coisas relacionadas ao trabalho de escritório estão localizadas na pasta Office.

Se você tem alguma opinião, algo que precisa ser esclarecido ou alguma sugestão, por favor, adicione na seção de comentários e por favor não se esqueça de curtir e compartilhar. Espero que todos gostem :)

* * *

Tem um amigo que precisa ler isso ? Mande a news para ele! [!\[\]\(919a2cb85b99741a73c0c31a427236a8_img.jpg\)](#)

Compartilhar Conteúdo

Encaminharam este conteúdo para você? Siga meu canal por aqui

Seguir meu canal

Quer saber mais sobre o dia a dia de um DEV? me segue no instagram

Quem é Desenvolvedor BR?

Me chamo Allan Monteiro, tenho 23 anos, moro no Rio de Janeiro, Brasil.

Eu sou um engenheiro de front-end, atualmente trabalhando com React, Next, JS, TS.

Se quiser conversar mais, fique à vontade para entrar em contato comigo.

Você pode verificar meu portfólio aqui:

allanmont.vercel.app

Programação & Desenvolvimento

#desenvolvedor

#programacao

#git

Publicação Anterior

Próxima Publicação

Outras publicações de desenvBR

Você sabe auditar a qualidade do seu código?? esse APP lhe ajuda ;)

Por Allan Monteiro

Hoje vim falar a vocês sobre um site e uma extensão que estou utilizando aos projetos, nela é possível verificar diversas métricas. falarei mais sobre isso ao longo do post

Git: Fluxo de trabalho prático

Por Allan Monteiro

Fluxo prático com os principais comandos do GIT

Git Workflow: o que é e seus principais tipos

Por Allan Monteiro

O Git é um projeto de código aberto maduro e com manutenção ativa desenvolvido em 2005 por Linus Torvalds, o famoso criador do kernel do sistema operacional Linux. Um número impressionante de projetos de software depende do Git para...

Resumo de HTML

Por Allan Monteiro

HTML (Linguagem de Marcação de Hipertexto) é o código que você usa para estruturar uma página web e seu conteúdo. Por exemplo, o conteúdo pode ser estruturado em parágrafos, em uma lista com marcadores ou usando imagens e tabelas...

Material-UI agora é MUI

Por Allan Monteiro

Você conhece React.js? Se sim, então você certamente conhece o Material-UI - uma das populares bibliotecas de componentes React. Devo dizer que Material-UI é a melhor biblioteca de uso geral do mercado. Ele segue as diretrizes,...

↓ [Ver mais publicações](#)

© 2022 **desenvBR**

Conheça os [termos de uso](#) e a [política de privacidade](#).

[Escreva com a Pingback](#)

